Chapter 6

Circuit Optimization

6.1 Introduction

Fast analog integration in deep sub-micron CMOS technologies has become a very important issue due to its growing economic impact. Since the technology roadmap predicts a fast scaling-down of the transistor's minimum channel lengths from 0.18μm in Year 2000 up to 0.03μm in Year 2014, the need for fast redesigns for different technologies of existing building blocks becomes crucial in the IC industry. Therefore, the design effort of a given block for a different technology requires the work of an expert analog designer to provide, all the equations containing the knowledge of the adopted topologies. On the other hand, the analog expert should also provide the design criteria for the optimization of these blocks [24-25,53-57].

As the demand for mixed-mode integrated circuits increases, the design of analog circuits such as operational transconductance amplifiers in CMOS technology becomes more critical. In circuit design optimization, a circuit and its performance specifications are given and the goal is to automatically determine the device sizes in order to meet the given performance specifications while minimizing a cost function, such as a weighted sum of the active area or power dissipation. The various methods for optimal determination of the component values and transistor dimensions for CMOS OTAs are introduced.

6.2 Optimization Methods

Existing approaches of automatic circuit sizing are broadly classified into four main categories [24-25]. These approaches are-

i) Classical optimization Methods

ii) Knowledge-Based Methods

iii) Global Optimization Methods

- Convex Optimization Methods

- Geometric Programming Methods.

iv) Designer-driven Stochastic Multi-objective Optimization Method.

The main disadvantage of the classical optimization methods is they only find locally optimal designs. This means that the design is at least as good as neighboring designs, i.e., small variations of any of the design parameters results in a worse (or infeasible) design. The knowledge – based methods find a locally optimal design (or, even just a "good" or "reasonable" design) instead of a globally optimal design. The final design depends on the initial design chosen and the algorithm parameters. The global optimization methods are that they are extremely slow, with computation growing exponentially with problem size, and cannot (in practice) guarantee a globally optimal solution. Convex optimization (*Geometric Programming*) methods can solve large problems, with thousands of variables and tens of thousands of constraints, very efficiently (in minutes on a small workstation). The other main advantage is that the methods are truly global, i.e., the global solution is always found, regardless of the starting point (which, indeed, need not be feasible) and infeasibility is unambiguously detected [24.58]. Designer-driven Stochastic Multi-objective Optimization Method uses

designer knowledge and requires only approximate equations. It gives much more accurate optimization with no added computational cost [33].

From this it seems that the convex optimization (*Geometric Programming*) method is the most suitable for solving circuit-sizing problems. But, it has one drawback that it needs expert designer knowledge to introduce the constraints in a special form. The method, which is presented, can be applied to a wide variety of amplifier architectures. In this thesis, the method is applied to a specific CMOS OTA.

6.2.1 Geometric Programming (GP)

These are special optimization problems in which the objective and constraint functions are all convex. While the theoretical properties of convex optimization problems have been appreciated for many years, the advantages in practice are only beginning to be appreciated now. The main reason is the development of extremely powerful interior-point methods for general convex optimization problems in the last five years [54].

Advantage - These methods can solve large problems, with thousands of variables and tens of thousands of constraints, very efficiently (in minutes on a small workstation). Problems involving tens of variables and hundreds of constraints are considered small, and can be solved on a small current workstation in less than one second. The extreme efficiency of these methods is one of their great advantages. The other main advantage is that the methods are truly global, i.e., the global solution is always found, regardless of the starting point (which, indeed, need not be feasible).

Infeasibility is unambiguously detected, i.e., if the methods do not produce a feasible point they produce a certificate that proves the problem is infeasible. Also, the stopping criteria are completely nonheuristic: at each iteration a lower bound on the achievable performance is given.

Disadvantages - One of the disadvantages is that the types of problems, performance specifications, and objectives that can be handled are far more restricted than any of the methods described above. This is the price that is paid for the advantages of extreme efficiency and global solutions [24].

A geometric program (GP) is a type of mathematical optimization problem characterized by objective and constraint functions that have a special form. The basic approach is to attempt to express a practical problem, such as an engineering analysis or design problem, in GP format. In the best case, this formulation is exact; when this isn't possible, we settle for an approximate formulation. Formulating a practical problem as a GP is called GP modeling. If we succeed at GP modeling, we have an effective and reliable method for solving the practical problem.

6.2.1.1 Basic Geometric Programming

Monomials and Posynomials

Let $x_1, x_2, \ldots x_n$ denote n real positive variables, and $x = (x_1, x_2 \ldots x_n)$ a vector with components x_i. A real valued function f of x, with the form

$$f(x) = cx_1^{a1} x_2^{a2} \ldots\ldots x_n^{an} \qquad (6.1)$$

where $c > 0$ and $ai \in R$ is called a monomial function, or more informally, a monomial (of the variables $x_1, x_2 \ldots x_n$). We refer to the constant c as the coefficient of the

monomial, and we refer to the constants a_i as the exponents of the monomial. . As an example, $9.5x_1x_2^{-0.1}$ is a monomial of the variables x1 and x2, with coefficient 9.5 and x_2-exponent -0.1. Any positive constant is a monomial, as is any variable. Monomials are closed under multiplication and division: if f and g are both monomials then so are fg and f/g. (This includes scaling by any positive constant.) A monomial raised to any power is also a monomial .

The 'monomial' will refer to the definition given above, in which the coefficient can be any positive number, and the exponents can be any real numbers, including negative and fractional.

A sum of one or more monomials, i.e., a function of the form :

$$f(x)=\sum_{k=1}^{K} c_k x_1^{a1k} x_2^{a2k} \ldots\ldots\ldots x_n^{ank} \qquad (6.2)$$

where $c_k > 0$, is called a posynomial function or, more simply, a posynomial (with K terms having the variables $x_1 \ldots$ xn). The term 'posynomial' is meant to suggest a combination of 'positive' and 'polynomial'. Any monomial is also a posynomial. Posynomials are closed under addition, multiplication, and positive scaling. Posynomials can be divided by monomials (with the result also a posynomial): If f is a posynomial and g is a monomial, then f/g is a posynomial. If Υ is a nonnegative integer and f is a posynomial, then f^{Υ} always makes sense and is a posynomial (since it is the product of Υ posynomials).

Let us give a few examples. Suppose x, y, and z are (positive) variables.

The functions (or expressions) $10.1x, 7xy\sqrt{z}, 5.8x^2yz^{-1}$ are monomials (hence, also posynomials). The functions $11.4xy/z, 6(xy+yz)^2, 7x+8y/xy$ are posynomials but not monomials [24,54].

6.2.1.2. Standard Form Geometric Program

A geometric program (GP) is an optimization problem of the form

minimize $f_0(x)$

$$\text{subject to } \begin{array}{l} f_i(x) \le 1, i = 1, \ldots\ldots, m, \\ g_i(x) = 1, i = 1, \ldots\ldots, p, \\ x_i > 0, i = 1, \ldots\ldots, n. \end{array} \qquad (6.3)$$

where fi are posynomial functions, g_i are monomials, and Xi are the optimization variables. (There is an implicit constraint that the variables are positive, i.e., $Xi > 0$.) We refer to the problem (6.3) as a geometric program in standard form. In a standard form GP, the objective must be posynomial (and it must be minimized); the equality constraints can only have the form of a monomial equal to one, and the inequality constraints can only have the form of a posynomial less than or equal to one [24,54].

6.2.1.3 Geometric Programming in Convex Form

The main trick to solving a GP efficiently is to convert it to a nonlinear but convex optimization problem, i.e., a problem with convex objective and inequality constraint functions, and linear equality constraints. Efficient solution methods for general convex optimization problems are well developed. This results in the problem

minimize $\log f_0(e^y)$

subject to $\log f_i(e^y) \le 0, i = 1, \ldots\ldots, m,$

$$\log g_i(e^y) = 0, i = 1, \ldots\ldots, p \qquad (6.4)$$

with variable y, where $y_i = \log x_i$. Here we use notation e^y, where y is a vector, to mean component wise exponentiation: $(e^y)_i = e^{y_i}$. The transformed version, given in relation (6.4), is the so-called convex form of the geometric program (6.3). Unlike the original GP, this convex form can be solved very efficiently [54].

Advantages of convex optimization are as follows:

These methods can solve large problems, with thousands of variables and tens of thousands of constraints, very efficiently. These methods are truly global, i.e., the global solution is always found, regardless of the starting point. Infeasibility is unambiguously detected, i.e., if the methods do not produce a feasible point they produce a certificate that proves the problem is infeasible. Also, the stopping criteria are completely nonheuristic, i.e. at each iteration point a lower bound on the achievable performance is given.